(2)

NASA Contractor Report 182048
ICASE Report No. 90–37

ICASE

STABLE BOUNDARY CONDITIONS FOR CARTESIAN GRID CALCULATIONS

M. J. Berger
R. J. LeVeque

Contract No. NAS1–18605
May 1990

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23665–5225

Operated by the Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

Stable Boundary Conditions for Cartesian Grid Calculations

M. J. Berger[1]
Courant Institute
251 Mercer St.
New York University
New York, NY 10012

and

R. J. LeVeque[1]
Department of Mathematics
University of Washington
Seattle, WA 98195

ABSTRACT

We solve the inviscid Euler equations in complicated geometries using a Cartesian grid. This requires solid wall boundary conditions in the irregular grid cells near the boundary. Since these cells may be orders of magnitude smaller than the regular grid cells, stability is a primary concern. We present a new approach to this problem and illustrate its use.

[1]The authors were supported in part by NSF Grants ASC-8858101 and DMS-8657319, AFOSR Grant 86-0148, and DOE Grant DE-FG02-88ER25053. This work was also supported in part by the National Aeronautics and Space Administration under NASA Contract No. NAS1-18605 while the authors were in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23665.

1 Introduction

In previous work [1], [6], [7], we have described a Cartesian grid method for the inviscid Euler equations in arbitrary geometries. There are many advantages to be gained from this approach. Grid generation is simplified, since we avoid the use of (possibly multiblock) body-fitted grids, and we can use high resolution, highly efficient solvers on regular grids over the bulk of the domain. This has led to renewed interest in Cartesian grids in recent years, *e.g.*, [3], [10]. One of the difficulties with Cartesian grids is that they give insufficient resolution in certain regions such as leading edges. This can now be overcome by Cartesian adaptive mesh refinement [1], [2].

The principal remaining difficulty in this approach is due to the essentially arbitrary way that a Cartesian grid intersects the boundaries of the computational domain. In particular, a solid wall boundary cutting through the grid creates irregular cells that may be orders of magnitude smaller than the regular cells away from the boundary. For these irregular cells, special difference equations are needed that maintain stability and accuracy, and satisfy the solid wall boundary conditions of no normal flow.

In this work, we present an improved method for the small boundary cells. We use an explicit, finite volume formulation that computes fluxes at cell edges on the regular part of the domain. We would like to define fluxes at the edges of the irregular cells in such a way that the method is stable even with very small boundary cells, using a time step based on the regular grid cells away from the boundary. The CFL condition requires that the numerical method allow information to propagate at least as quickly as the underlying differential equation. In the present context this means that we must define fluxes at the sides of our irregular cells based on more than just the neighboring cell values.

In our previous work, we have used a wave propagation approach in defining these fluxes. Here we propose an alternative method that has some advantages over the wave propagation approach. In particular, the wave propagation method is subject to intermittent instabilities due to two-dimensional effects that are not clearly understood. The new method has a cancellation property in two dimensions that appears to give better stability properties. Moreover, the computational geometry is simplified in the new approach. The fluxes are defined in terms of weighted averages of nearby cell values. These weights may be calculated as a preprocessing step on any fixed grid and need not be repeatedly calculated. In the previous approach the weights depend on the flow variables and a certain amount of computational geometry was required

near the boundary in every time step.

We consider the inviscid Euler equations of gas dynamics in two space dimensions,

$$u_t + f(u)_x + g(u)_y = 0 \tag{1.1}$$

where

$$u = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho E \end{bmatrix}, \quad f(u) = \begin{bmatrix} \rho u_1 \\ \rho u_1^2 + p \\ \rho u_1 u_2 \\ u_1(\rho E + p) \end{bmatrix}, \qquad g(u) = \begin{bmatrix} \rho u_2 \\ \rho u_1 u_2 \\ \rho u_2^2 + p \\ u_2(\rho E + p) \end{bmatrix}. \tag{1.2}$$

Here (u_1, u_2) represents the velocity, E is the total energy per unit mass, and p is the pressure, which is related to the other variables by the equation of state. We assume a γ-law gas, so that

$$p = (\gamma - 1)(\rho E - \frac{1}{2}\rho(u_1^2 + u_2^2)). \tag{1.3}$$

At a solid wall boundary we require that the component of velocity normal to the wall be zero.

In one space dimension the system reduces to

$$u_t + f(u)_x = 0 \tag{1.4}$$

where $u = (\rho, \rho v, \rho E)$ and $f(u) = (\rho v, \rho v^2 + p, v(\rho E + p))$, with $v = u_1$ the velocity. The boundary conditions become $v = 0$ at a solid wall.

2 A one-dimensional example

In order to illustrate this approach we begin with a one-dimensional model problem, the one-dimensional Euler equations for $x > 0$ with a solid wall at $x = 0$. We take a grid with cell interfaces at the points

$$\begin{aligned} x_0 &= 0 \\ x_1 &= h' \\ x_j &= h' + jh \quad \text{for } j = 2, 3, \ldots. \end{aligned}$$

Here h is a uniform grid spacing and $h' \leq h$. the grid is uniform except for one small cell near the boundary (see Figure 1). We use a conservative method in the form

$$U_j^{n+1} = U_j^n - \frac{k}{h_j}[F_{j+1}^n - F_j^n], \quad j = 0, 1, \ldots. \tag{2.5}$$

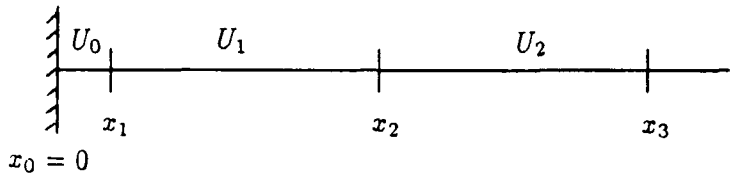

Figure 1: One dimensional grid with one irregular cell adjacent to the wall.

Here h_j is the width of the jth cell, so in our case we have $h_0 = h'$ and $h_j = h$ for $j > 0$.

For simplicity we restrict our attention to Godunov's method on the regular portion of the grid, although the ideas we propose can be extended to higher order methods as well. In Godunov's method we take

$$F_j^n = f(u^*(U_{j-1}^n, U_j^n)) \tag{2.6}$$

where $u^*(u^L, u^R)$ represents the solution to the Riemann problem with left and right states u^L and u^R, evaluated along $x/t = 0$. Although a rigorous stability proof is not available for systems of equations, in practice this method is always stable provided the CFL condition

$$\left| \frac{k\lambda_{\max}}{h} \right| \leq 1 \tag{2.7}$$

is satisfied, where $\lambda_{\max}$ is the maximum wave speed. We will assume that our time step k is chosen so that the condition (2.7) is satisfied relative to the uniform h. We will use the flux (2.6) for $j = 2, 3, \ldots$, *i.e.*, at all interfaces where the cell on both sides is regular. Our task is to define fluxes F_j^n for $j = 0, 1$ so that we maintain stability (and accuracy) with this time step even if $h' \ll h$.

First suppose $h' = h$. Then we can use the Godunov flux (2.6) also at $j = 1$. At the wall we use the well-known observation that the solution to the boundary value problem can be obtained by ignoring the wall and extending the computational domain to the whole line $-\infty < x < \infty$, if we take data $u_0(x)$ for $x < 0$ equal to

$$\begin{aligned} \rho(x,0) &= \rho(-x,0) \\ v(x,0) &= -v(-x,0) \qquad \text{for } x < 0 \\ p(x,0) &= p(-x,0). \end{aligned}$$

We will denote this "reflection" of the data (in which the velocity is negated) by the operator $\mathcal{R}$, so that for shorthand we can write

$$u(x,0) = \mathcal{R}(u(-x,0)) \quad \text{for } x < 0.$$

With this extended data, the solution continues to satisfy $u(x,t) = \mathcal{R}(u(-x,t))$ also for $t > 0$ and in particular the boundary condition $u(0,t) = 0$ is automatically satisfied. This suggests that we obtain a flux at the wall by solving a Riemann problem with left and right states

$$u^L = \mathcal{R}(U_0), \qquad u^R = U_0$$

in each time step to obtain

$$F_0 = f(u^*(\mathcal{R}(U_0), u_0).$$

(For brevity we will leave off the superscript n in general.) Note that the density and energy components of this flux will be zero since the velocity component of u^* is zero at the wall. There will only be a momentum flux at the wall due to the pressure there, as expected physically.

If $h' < h$ we could attempt to use this same formula to define F_0 but we would find that it is unstable unless the CFL condition

$$\left| \frac{k\lambda_{\max}}{h'} \right| \leq 1 \tag{2.8}$$

is satisfied. This will place an unreasonable restriction on k if $h' \ll h$.

This instability is caused by the fact that the boundary flux F_0 is based on the data U_0 alone. If the CFL condition (2.8) is satisfied, then it is only this data that affects the flux at the wall over the time step. However, when (2.8) is violated the value U_1 should also affect the flux at the wall, and ignoring this effect leads to instability.

In a "large time step" approach we increase the stencil of the method, meaning we allow more data points to come into the computation of each flux, and hence retain stability. One way to achieve this is by a wave propagation approach. The solution of the Riemann problem at each cell interface consists of three waves propagating away from the interface. If (2.8) is satisfied then these waves remain in the cells bordering the interface during the entire time step and hence affect the solution only in these cells. If (2.8) is violated then the waves may affect cells further away. Implementing Godunov's method in terms of this wave propagation approach, allowing waves to affect more than just the adjacent cell, gives a large time step generalization that remains stable for much larger time steps[5]. In the present context this allows us to reduce h' without reducing the time step k. Waves from the boundary Riemann problem cross the interface at x_1 and affect U_1 as well as U_0. Waves from the interface at x_1 may reach the boundary. These waves reflect off the boundary and the reflected

wave affects the value U_0 and perhaps also U_1 if the reflected wave reaches the cell interface at x_1 during the time step.

A more detailed description of this procedure may be found in [7]. A natural extension to two space dimensions gives one method to deal with small cells near the boundary, as described in [1], [6], [7]. In one dimension this works very well but in two dimensions occasional stability problems have still been observed due to multidimensional effects.

The new approach. Our new approach to the small cell problem can also be illustrated with the one-dimensional problem described above. We again use the method (2.5) with Godunov fluxes (2.6) for $j = 2, 3, \ldots$. For $j = 0$ and $j = 1$ we define fluxes in a similar manner but with a different choice of states u^L and u^R in the Riemann solver. Recall that in a naive attempt to use Godunov's method regardless of the size h' of the small cell we would take left and right states

$$u_0^L = \mathcal{R}(U_0) \qquad u_0^R = U_0 \tag{2.9}$$

$$u_1^L = U_0 \qquad u_1^R = U_1. \tag{2.10}$$

To maintain stability when h' is small, we need to allow data from additional grid cells to affect the left and right states at each of these interfaces. Recall that the method is assumed to be stable with our choice of k and h on the regular portion of the grid. This suggests that we should define u_j^L by taking the average value of U over an interval of length h to the left of the interface x_j and define u_j^R by taking the average value of U over an interval of length h to the right of x_j.

For example, at $x_0 = 0$ (the wall) we set

$$u_0^R = \frac{1}{h}(h'U_0 + (h - h')U_1) \tag{2.11}$$

If we view the grid values as defining a piecewise constant function with values U_j in the jth cell, then (2.11) is the average value of this function over the interval $0 \le x \le h$. Note that if $h' = h$ (the grid is completely regular) then (2.11) reduces to $u_0^R = U_0$ as expected for Godunov's method. Recall that in Godunov's method we take $u_0^L = \mathcal{R}(U_0) = \mathcal{R}(u_0^R)$ to impose the boundary condition $v(0,t) = 0$. This suggests that more generally we take

$$u_0^L = \mathcal{R}(u_0^R) \tag{2.12}$$

where u_0^R is defined by (2.11). We then use the Godunov flux

$$F_0 = f(u^*(u_0^L, u_0^R)) \tag{2.13}$$

as the flux at the wall. Using (2.12) guarantees that there will be no flux of mass or energy through the wall and hence that the method is conservative.

To define the left and right states at x_1 we again construct intervals of length h to either side of this point and average the piecewise constant function defined by U over these intervals. To the right of x_1 lies a regular cell of length h and so

$$u_1^R = U_1. \tag{2.14}$$

To the left of x_1 an interval of length h extends beyond the wall (assuming $h' < h$). Beyond the wall we assume that U takes the value u_0^L given by (2.12). A weighted average of this value and U_0 gives u_1^L:

$$u_1^L = \frac{1}{h}(h'U_0 + (h - h')u_0^L). \tag{2.15}$$

The flux f_1 is then defined by

$$F_1 = f(u^*(u_1^L, u_1^R)). \tag{2.16}$$

Again, if $h' = h$ this reduces to the standard Godunov flux.

This method remains stable even when $h' \ll h$. To see why this should be so, consider the formula (2.5) for $j = 0$ where $h_j = h'$. It is the division by h' that may cause stability problems unless the fluxes F_0 and F_1 themselves agree to $O(h')$ as $h' \to 0$. The Godunov fluxes based on (2.9), (2.10) do not have this property. However, our proposed fluxes (2.13) and (2.16) do have this property, since inspection of the formulas (2.11), (2.12), (2.14), and (2.15) shows that $u_1^L = u_0^L + O(h')$ and $u_0^R = u_1^R + O(h')$ as $h' \to 0$. Since the flux function $f(u^*(u^L, u^R))$ is a Lipschitz continuous function of u^L and u^R, it follows that $F_1 - F_0 = O(h')$ as $h' \to 0$ and there is at least a chance that the method remains stable for arbitrary $h' \ll h$. Numerical experiments show that this is indeed the case (although it is possible to contrive examples, such as a strong rarefaction wave originating at this irregularity, where the results are not very accurate).

3 Boundary conditions in two dimensions

Turning now to the two-dimensional problem, we will give a brief description of how the idea described above extends to handle the small cell problem.

Consider the portion of the boundary shown in Figure 2a and a typical boundary cell (i, j). The formula for updating the value U_{ij} is the two-dimensional analog of (2.5),

$$U_{ij}^{n+1} = U_{ij} - \frac{k}{A_{ij}}[F_{i+1,j} - F_{ij} + G_{i,j+1} - G_{ij} + H_{ij}]. \tag{3.17}$$

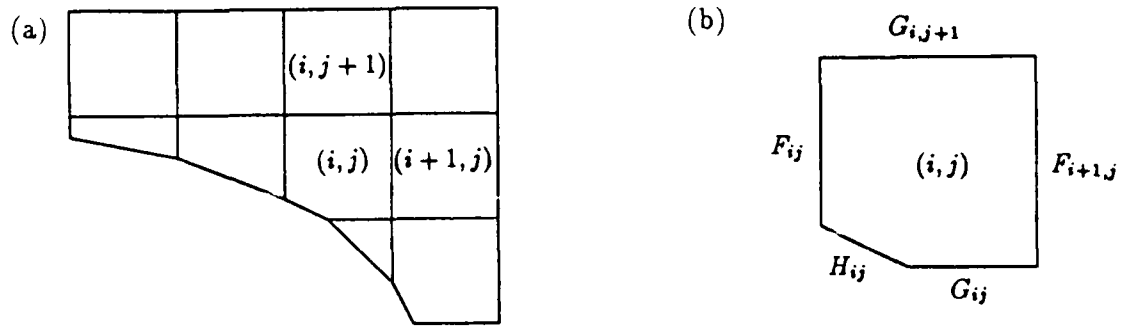

Figure 2: (a) The Cartesian grid near the boundary. (b) Blow up of cell (i,j) showing the location of fluxes.

The fluxes F, G, and H represent flux per unit time through the corresponding side of the grid cell (see Figure 2b) and A_{ij} is the area of the cell. If any of the sides are missing, the corresponding flux is zero.

On regular grid cells, $H_{ij} = 0$ and the fluxes F and G might be defined by an extension of the Godunov method, setting

$$F_{ij} = hf(u^*(U_{i-1,j}, U_{ij})), \qquad G_{ij} = hg(u^*(U_{i,j-1}, U_{ij})). \tag{3.18}$$

Here u^* represents the solution to the appropriate one-dimensional Riemann problem in the x or y direction. Note that the fluxes include the factor h, the length of each side, to give a flux per unit time across the side.

It is the denominator A_{ij} in (3.17) that causes trouble when the cell is very small. We again assume the method is stable on the regular portion of the grid, where $A_{ij} = h^2$. To maintain stability we need to insure that our formulas for the fluxes cause the total flux (the sum in brackets in (3.17)) to cancel to $O(A_{ij})$ as $A_{ij} \to 0$. This is only possible if the fluxes are computed via formulas that involve more than just the two cells bordering the cell side. We take an approach analogous to what we described above in one dimension.

Boundary fluxes. We begin by considering the boundary segment, where we must compute the flux H_{ij}. In two dimensions the solid wall boundary condition requires that the normal velocity at the wall be equal to zero. If we have some value u_{ij}^{in} representing the value of u just inside the wall, then we can obtain the flux H_{ij} by solving a one-dimensional Riemann problem in the direction normal to the wall, with left and right states

$$u_{ij}^L = \mathcal{R}(u_{ij}^{\text{in}}), \qquad u_{ij}^R = u_{ij}^{\text{in}}.$$

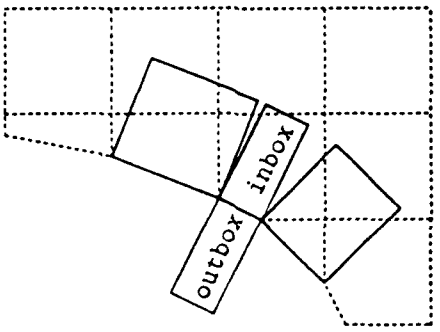

Figure 3: The inbox and outbox constructed from the boundary segment of cell (i, j), and the inbox for two neighboring cells.

The reflection operator $\mathcal{R}$ is now defined by negating the normal velocity component while leaving the tangential velocity component along with the density and pressure unchanged. The resulting Godunov flux is used for H_{ij}.

We obtain u_{ij}^{in} by a procedure analogous to the one-dimensional example. We construct a box extending a distance h away from the wall as shown in Figure 3. The box extending into the computational domain is called $\texttt{inbox}(i, j)$. The mirror image box outside the domain is called $\texttt{outbox}(i, j)$. We obtain the value u_{ij}^{in} by viewing the given data U as defining a piecewise constant function, constant in each grid cell, and setting u_{ij}^{in} to be the average value of this function over the region $\texttt{inbox}(i, j)$. In Figure 3 $\texttt{inbox}(i, j)$ would contain an area-weighted average of two grid values while the value for $\texttt{inbox}(i + 1, j)$ is based on four grid values. We think of the outbox as containing the value $u_{ij}^{\text{out}} = \mathcal{R}(u_{ij}^{\text{in}})$.

To find the weights needed to compute u_{ij}^{in} we must compute the intersection of the inbox with each nearby cell. This is easily accomplished with standard computational geometry routines. Note that for a given geometry and grid these weights need only be computed once at the beginning of the computation. They need not be recomputed in each time step.

Fluxes at other sides. We now consider the fluxes F and G along other sides of this cell. These are all computed by similar procedures, so to be specific we will consider the computation of F_{ij}, the flux on the left side of this cell.

To compute F_{ij} we solve two Riemann problems, one in some direction ξ with some data u_ξ^L, u_ξ^R and the other in the orthogonal direction η with data u_η^L, u_η^R. The choice of these directions and the data will be discussed in a moment. First we explain how these Riemann problem solutions are computed and used to define F_{ij}.

Figure 4 shows a typical vertical cell interface and two orthogonal directions ξ and

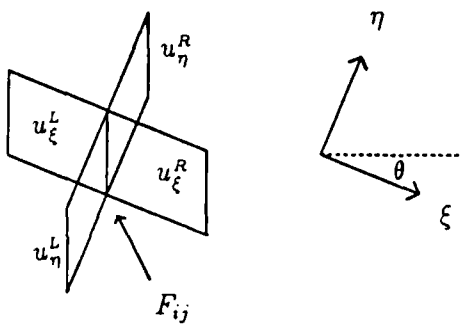

Figure 4: A vertical cell interface and the ξ- and η-directions.

η. Let θ be the angle that ξ is rotated from the x-direction ($\theta < 0$ in this example). Suppose we solve a one-dimensional Riemann problem in the ξ-direction with left and right states u_ξ^L, u_ξ^R to obtain the flux per unit length per unit time in the ξ-direction. (To do this we rotate the velocity cor ponents of u_ξ^L, u_ξ^R into ξ-η velocity components, solve the one-dimensional Riemann problem, and then rotate the resulting flux f back to x-y velocity components.) Call this resulting flux f_ξ.

Similarly, solving a one-dimensional Riemann problem in the η-direction with left and right states u_η^L, u_η^R gives f_η, the flux per unit length per unit time in the η-direction. The total flux across the vertical segment of length h' is then

$$F = h'(f_\xi \cos\theta - f_\eta \sin\theta). \tag{3.19}$$

This is the value we use for the flux F_{ij}.

This same approach has been used by others (*e.g.*, [4], [8], [9]) to define multi-dimensional upwind methods. In these methods the directions ξ and η are chosen based on the local flow in an attempt to use physically meaningful directions in place of the artificial coordinate directions. For example, the direction of the velocity or the pressure gradient might be used to define ξ. In our application we are only considering cells adjacent to the boundary and the relevant directions are the directions tangential and normal to the wall. We choose ξ to be the direction tangential to the wall in one of the two cells bordering this interface. Since our primary concern is to maintain stability in very small cells, we choose the smaller of the two adjacent cells to define this direction. This will lead to cancellation of fluxes in tiny cells in the same manner as previously seen in the one-dimensional example. The η-direction is normal to the ξ-direction.

Tangential boxes. We must still specify the data for these tangential and normal Riemann problems. We first consider the tangential problem. We use an approach

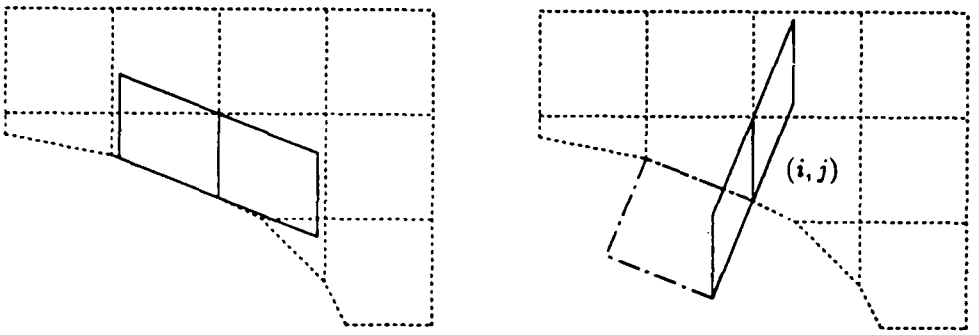

Figure 5: (a) Tangential boxes constructed from the cell interface. (b) Normal boxes from the cell interface and outbox$(i-1,j)$.

similar to the specification of data in an inbox described above. From the interface we construct boxes that extend a distance h in the ξ-direction. Figure 5a shows an example. The data u_ξ^L, u_ξ^R is obtained by an area-weighted average of the values in each cell the box overlaps. In our current implementation we assume the wall is convex, so that these boxes lie entirely within the computational domain. Each box overlaps at most two grid cells and the weights are easily calculated. Since the directions ξ and η and the resulting boxes depend only on the geometry, not on the flow variables, these weights can again be calculated once and for all as a preprocessing step.

Normal boxes. Figure 5b shows the normal boxes in the η-direction. The box in the outward direction does not hit the boundary and overlaps at most two regular cells, so u_η^R is calculated as an area-weighted average of these cell values. The other box may extend beyond the boundary. If so, the portion lying outside the computational domain lies in one or more outboxes, the artificial cells created in the process of computing the boundary flux H_{ij} described above. Figure 5b shows a simple example where the normal box intersects only one cell $(i-1,j)$ and outbox$(i-1,j)$. More generally the normal box might intersect two cells and their outboxes, as happens for example when we compute the flux $F_{i+1,j}$ which involves cells (i,j) and $(i,j-1)$. Moreover the two outboxes will in general overlap due to the convexity of the region. We again use area-weighted averaging over the four cells in question, weighting the values U_{ij}, $U_{i,j-1}$, u_{ij}^{out}, $u_{i,j-1}^{\text{out}}$ by the areas of intersection and then dividing by the sum of all these areas.

Cancellation. Although we will not present the details here, it can be shown that this way of defining fluxes leads to the desired cancellation of fluxes in very small cells. The values u_ξ^L computed at each of the three sides of a very small triangular

cell are nearly the same because of our construction. They differ by only $O(A_{ij})$ as $A_{ij} \to 0$. The same is true of each of the other values u_ξ^R, u_η^L, u_η^R and so by Lipschitz continuity of the fluxes F, G and H we obtain the required cancellation. Numerical results show stability even when A_{ij} is many orders of magnitude less than h^2.

Higher order methods The method (3.17) with fluxes (3.18) is only first order accurate and is highly dissipative. In our previous work we used the wave propagation boundary conditions together with a high resolution method away from the boundary and obtained reasonable results (*e.g.*, [1]). The new boundary conditions can also be applied in conjunction with a high resolution method and gives similar results. Moreover, with our new formulation it appears to be easier to improve the accuracy of the boundary conditions, allowing us to obtain higher order accuracy overall. The main idea is to introduce slopes in each cell and use piecewise linear approximations in place of piecewise constants to define the fluxes. Near the boundary we can easily estimate slopes in the tangential direction along the wall by differencing values in the inboxes that we have defined above.

These improvements are still being investigated and will be reported in detail elsewhere. Here we will only compare results obtained with the method as we have described it and results obtained using the same interior method with the wave propagation boundary conditions described in earlier papers.

4 Numerical results

We show one representative test case, a supersonic shock going around an expansion corner. We also show the steady state solution obtained at large times. The exact rarefaction wave solution is a simple wave and can be computed following Section 6.17 of Whitham[11], for example.

The geometry we use is the rectangle $[0, 1.32] \times [0, 0.8]$ with a solid wall at

$$y = \begin{cases} 0.3 & x \le 0.1 \\ 0.3(1 - (x - 0.1)^2) & 0.1 \le x \le 0.7 \\ 0.192 - 0.36(x - 0.7) & 0.7 \quad \le x \le 1.32. \end{cases}$$

The initial conditions consist of a Mach 2.31 shock at $x = 0.06$ with left and right states

$$\rho^L = 5.1432, \quad u_1^L = 2.04511, \quad u_2^L = 0, \quad p^L = 9.04545$$

and

$$\rho^R = 1.4, \quad u_1^R = 0, \quad u_2^R = 0, \quad p^R = 1.0.$$

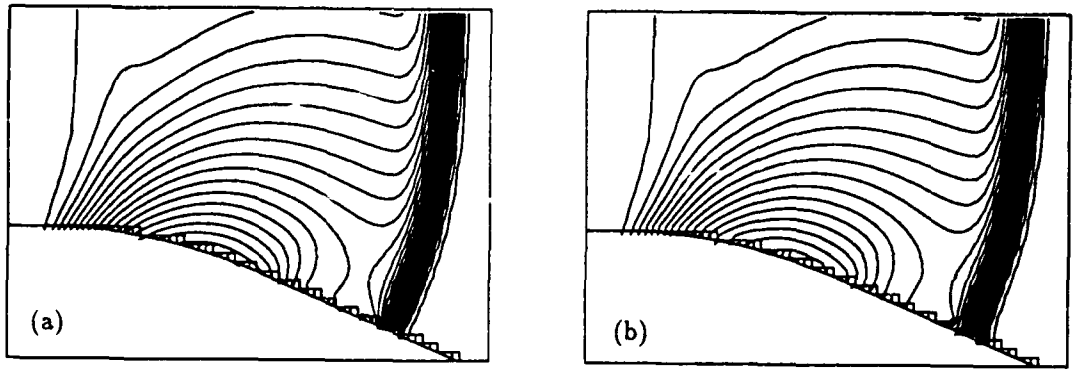

Figure 6: Shock propagation results at $t = 0.4$ (a) Using the wave propagation boundary conditions. (b) Using the new boundary conditions.

We take $h = 0.02$ (66×40 grid) and a time step $k = 0.002$. This corresponds to a Courant number of roughly 0.37 relative to the regular cells with area h^2. For the crude form of Godunov's method used here, the stability restriction requires Courant number less than 0.5. The smallest cells near the boundary have an area roughly $10^{-3}h^2$.

Figure 6 shows numerical results at time $t = 0.4$, as the shock is rounding the corner. Results obtained with the wave propagation boundary conditions are shown in Figure 6a, while Figure 6b shows the results obtained with our new approach. These results are very similar. Slight discrepancies can be seen near the wall just around the shock. For this problem both sets of boundary conditions worked well. We have also performed tests on other problems where the wave propagation method shows instabilities and have observed no such difficulties with the new method.

Figure 7 shows the steady state results obtained after many iterations of the time dependent code (no attempt has been made so far to accelerate convergence for steady state solutions). We only show the results with our new boundary conditions. The wave-propagation boundary conditions give very similar results. We use a coarser grid than in the previous example ($h = 0.4$) in order to demonstrate that we achieve reasonable accuracy along the boundary even with a relatively coarse piecewise linear representation of the boundary. We also use a larger computational domain, $[0, 2] \times [0, 1.6]$ to minimize the impact of the far-field boundaries. The true solution is a rarefaction wave originating from the portion of the boundary with nonzero curvature. In the exact solution the contour lines would be straight lines. Our results are contaminated by effects from the far-field boundary.

Near the solid wall the contour lines appear to show a boundary layer. This is an artifact of the graphics routine, which assumes the data is on a uniform grid at cell

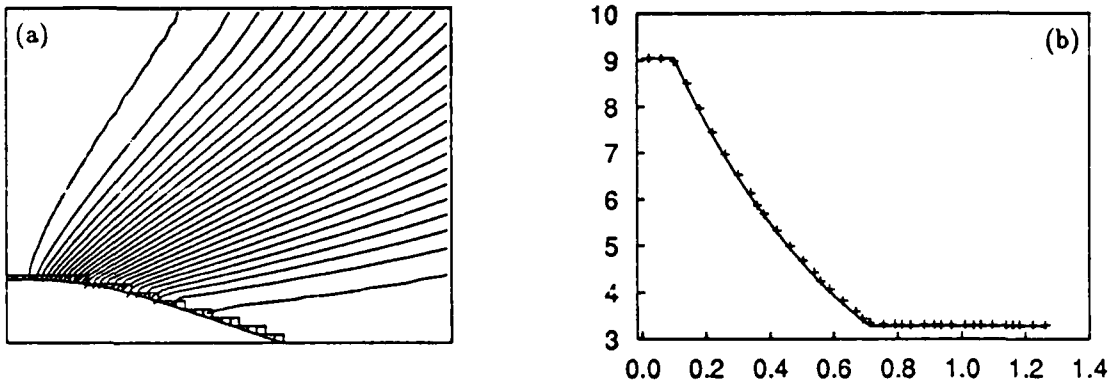

Figure 7: Steady state results. (a) Pressure contours. (b) Pressure along the wall. The solid line is the exact solution. + marks are the numerical solution.

centers. Our data near the boundary should be viewed as an approximation to the pointwise value at the center of mass of the irregular cell, not at the center of the full Cartesian cell.

In order to examine the accuracy at the wall, Figure 7b shows plots of the pressure along the boundary, plotted against arclength. To obtain a boundary pressure, the cell value U_{ij} and the reflected value $\mathcal{R}(U_{ij})$ are used to solve the one-dimensional Riemann problem normal to the wall in each irregular cell. The resulting pressure p^* is used as the boundary pressure. Figure 7b shows these results and also the exact solution (to machine precision) calculated using the theory of [11].

In more complicated computations we use adaptive grid refinement to obtain high resolution results with minimal effort. The boundary conditions described here can also be used in conjunction with the adaptive Cartesian grid code described in [1] and [2].

Acknowledgments. It is a pleasure to acknowledge several stimulating conversations with John Bell and Phil Colella.

References

[1] M. Berger and R. J. LeVeque, *An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries.* AIAA paper AIAA-89-1930, 1989.

[2] M. J. Berger and P. Colella, *Local adaptive mesh refinement for shock hydrodynamics*, J. Comput. Phys., 82 (1989), pp. 64–84.

[3] H. H. D. Clarke and M. Salas, *Euler calculations for multieliment airfoils using Cartesian grids.* AIAA Paper 85-0291, 1985.

[4] S. F. Davis, *A rotationally biased upwind difference scheme for the Euler equations*, J. Comput. Phys., 56 (1984), pp. 65–92.

[5] R. J. LeVeque, *A large time step generalization of Godunov's method for systems of conservation laws*, SIAM J. Num. Anal., 22 (1985), pp. 1051–1073.

[6] ——, *Cartesian grid methods for flow in irregular regions*, in Num. Meth. Fl. Dyn. III, K. W. Morton and M. J. Baines, eds., Clarendon Press, 1988, pp. 375–382.

[7] ——, *High resolution finite volume methods on arbitrary grids via wave propagation*, J. Comput. Phys., 78 (1988), pp. 36–63.

[8] D. W. Levy, K. G. Powell, and B. van Leer, *An implementation of a grid-independent upwind scheme for the Euler equations.* AIAA Paper 89-1931-CP, 1989.

[9] K. G. Powell and B. van Leer, *A genuinely multi-dimensional upwind cell-vertex scheme for the Euler equations.* AIAA Paper 89-0095, Reno, 1989.

[10] B. Wedan and J. South, *A method for solving the transonic full-potential equations for general configurations.* Proc. AIAA Computational Fluid Dynamics Conference, 1983.

[11] G. Whitham, *Linear and Nonlinear Waves*, Wiley-Interscience, 1974.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-182048 ICASE Report No. 90-37		

4. Title and Subtitle	5. Report Date
STABLE BOUNDARY CONDITIONS FOR CARTESIAN GRID CALCULATIONS	May 1990
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
M. J. Berger R. J. LeVeque	90-37
	10. Work Unit No. 505-90-21-01

9. Performing Organization Name and Address	11. Contract or Grant No.
Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665-5225	NAS1-18605
	13. Type of Report and Period Covered Contractor Report

12. Sponsoring Agency Name and Address	14. Sponsoring Agency Code
National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	

15. Supplementary Notes

Langley Technical Monitor:
Richard W. Barnwell

Final Report

To appear in
Symposium on Computational Technology for Flight Vehicles, ed. by Ahmed Noor, 1990

16. Abstract

We solve the inviscid Euler equations in complicated geometries using a Cartesian grid. This requires solid wall boundary conditions in the irregular grid cells near the boundary. Since these cells may be orders of magnitude smaller than the regular grid cells, stability is a primary concern. We present a new approach to this problem and illustrate its use.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Boundary Conditions, Euler Equations, Cartesian Grid	64 - Numerical Analysis Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	16	A03

NASA-Langley, 1990